Remarks:

Reconsideration of the application is respectfully requested.

Claims 3 and 4 are presently pending in the application. As it is believed that the claims were patentable over the cited art in their original form, the claims have not been amended to overcome the references.

In item 6 of the above-identified Office Action, claims 3 and 4 were rejected as allegedly being indefinite under 35 U.S.C. § 112, first paragraph. More particularly, it was alleged in the Office Action that, since the Applicants argued in their response to the previous Office Action (the "**Response**") that **YOSHIDA** required "**a complicated cascade of adding units**", it was unclear as to what Applicants regarded to be their claimed "computation unit". More particularly, it was stated in the Office Action, in part:

> It is unclear to the Examiner how the claimed invention's arrangement of adding unit and computation unit differs from Yoshida's disclosed "complicated cascade of adding units". However Applicant has argued the claimed invention and the "complicated cascade of adding units" of Yoshida are different from each other yet the specification does not provide details to enable one of ordinary skill in the art to make/use a "computation unit for computing relative addresses" without using an adder as disclosed in Yoshida and as is well known in the art.

Applicants' respectfully disagree that the "computation unit" of Applicants' claims is not enabled under 35 U.S.C. § 112.

More particularly, MPEP § 2164.01 states, in part:

> Any analysis of whether a particular claim is supported by the **disclosure** in an application requires a determination **of whether that disclosure**, when filed, contained sufficient information regarding the subject matter of the claims **as to enable one skilled in the pertinent art to make and use the claimed invention**. [emphasis added by Applicants]

Additionally, MPEP § 2164.01(b) states, in part:

> As long as **the specification discloses at least one method for making and using the claimed invention that bears a reasonable correlation to the entire scope of the claim, then the enablement requirement of 35 U.S.C. 112 is satisfied.** [emphasis added by Applicants]

As can be seen from MPEP § 2164, enablement of a claim is determined from a review of Applicants' **specification**, and is not based on **arguments** made by an Applicant in a response. Further, the enablement requirement is met if the specification discloses **at least one method** for making and using the claimed invention. MPEP § 2164.01 additionally states, in part:

> A patent need not teach, and preferably omits, what is well known in the art.

See also, MPEP § 2164, which states, in part:

> Detailed procedures for making and using the invention may not be necessary if the description of the invention itself is sufficient to permit those skilled in the art to make and use the invention

---

The Office Action admits that a person of skill in the art, upon reading Applicants' specification, would know **at least one method** for making the computation unit of the claimed invention. More particularly, the Office Action states, on page 4:

> However Applicant has argued the claimed invention and the "complicated cascade of adding units" of Yoshida are different from each other yet the specification does not provide details to enable one of ordinary skill in the art to make/use a "computation unit for computing relative addresses" **without using an adder as disclosed in Yoshida and as is well known in the art.**

As such, using an adder in the computation unit (i.e., which **is at least one method of making** the claimed invention), is admittedly well known in the art and would be apparent to a person of ordinary skill. As such, Applicants' claims are **enabled** under 35 U.S.C. § 112, first paragraph, as shown by the requirements of MPEP § 2164.

Further, Applicants' believe that the **specification** (i.e., which is what determines enablement) includes examples that would enable a person of ordinary skill in the art to make Applicants' particularly claimed "computation unit for

computing relative addresses". For example, paragraph [0008] of the instant application states:

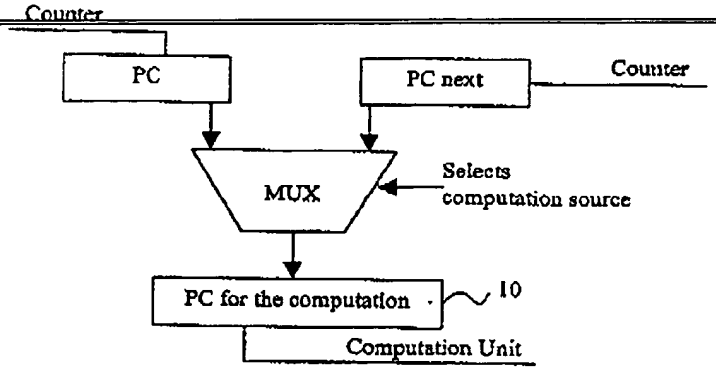
> ~~The present invention serves for developing a~~ microprocessor which can process different assembler codes. A great difficulty here is that, in the case of different assembler codes, the computation of relative addresses relates to different program counter definitions. For example, the relative addressing in the case of the JAVA byte code always relates to the current assembler instruction, in the case of ECO 2000 Assembler it always relates to the instruction counter reading that is pointing to the next assembler instruction to be executed. [emphasis added by Applicants]

Applicants' invention sets out to address the problems of the prior art, set forth in paragraph [0008] of the instant application. For example, paragraph [0030] of the instant application states:

> In all the figures of the drawing, sub-features and integral parts that correspond to one another bear the same reference symbol in each case. Referring now to the figures of the drawing in detail and first, particularly, to FIG. 1 thereof, there is shown a first embodiment of the invention in which two instruction counters PC, PCnext are provided in a microprocessor. The instruction counters PC, PCnext in each case contain an instruction counter reading belonging to a corresponding assembler code. One of the counters PC is consequently always pointing to a current program line (for example for JAVA byte code), while a further instruction counter PCnext is always pointing to the program line of a next assembler instruction (for example for ECO 2000 Assembler). The outputs of the two instruction counters PC, PCnext are connected to a multiplexer unit MUX, which, dependent on the assembler code to be processed at the respective time, connects one or the other instruction counter reading through to its output, which is connected to a computation unit 10 for the relative addresses. [emphasis added by Applicants]

Fig. 1 of the instant application is reproduced herebelow, for convenience.

FIG 1



See also, paragraph [0010] of the instant application, which states:

> With the foregoing and other objects in view there is provided, in accordance with the invention, a microprocessor for processing various assembler codes. The microprocessor contains a parameter designating a respective assembler code and, depending on how the parameter is set, a different relative addressing takes place. A plurality of program counters are provided and, dependent on the parameter, in each case **one of the program counters is active in a computation of relative addresses.** [emphasis added by Applicants]

As such, the computation unit of Applicants' claims is clearly enabled **by the specification**, such that a person of ordinary skill in the art could make it.

Note that, in the embodiment of Fig. 1, there is no additional adder between the MUX and the computation unit. Thus, in at least the embodiment of Fig. 1, there is no "**cascade of**

adders" as stated in Applicants' previous **Response**. Note also, in the embodiments of Figs. 2 - 3, the shown ADD and SUB boxes are present only to add or subtract, respectively, an instruction length to/from the selected instruction counter provided at the output of the MUX, as disclosed in paragraphs [0031] and [0032] of the instant application.

It is accordingly believed that the specification and the claims meet the requirements of 35 U.S.C. § 112, first paragraph.

Further, in item 8 of the Office Action, claims 3 and 4 were rejected under 35 U.S.C. § 103(a) as allegedly being obvious over U. S. Patent No. 5,854,913 to Goetz et al ("**GOETZ**"), in view of U. S. Patent No. 4,926,323 to Yoshida ("**YOSHIDA**") [sic], Mano and Kime, "Logic and Computer Design Fundamentals" ("**MANO**"), "The PowerPC Architecture", 1994 ("**POWERPC**") and K. Short, "Embedded Microprocessor Systems Design", 1998 ("**SYSTEMS DESIGN**").

Applicants respectfully traverse the above rejections.

First, Applicants note that the Office Action cites U. S. Patent No. **4,926,323** to **YOSHIDA**, against the instant claims. However, U. S. Patent No. **4,926,323** is <u>not</u> the **YOSHIDA** patent.

More particularly, U. S. Patent No. **4,926,323** is a patent to **Borar et al.**, while U. S. Patent No. **5,088,030** is to **YOSHIDA**. To further compound Applicants' confusion, **BORAR is only mentioned, and not even cited, in item 17 of the Office Action, with YOSHIDA**. Applicants' request clarification of the rejections of the instant application and request that, if any **different** rejections are made in the next Office Action **as a result of the erroneous naming of references in the instant Office Action**, that the next Office Action **not** be made final.

For purposes of responding to the instant Office Action, Applicants' have assumed that the patent meant in item 8 of the Office Action is U. S. Patent No. **5,088,030** to **YOSHIDA**, and not U. S. Patent No. **4,926,323** to **BORAR**.

Applicants' believe that the instant claims are patentable over the cited references, taken alone or in combination.

More particularly, Applicants' claim 3 recites a microprocessor for processing various assembler codes, comprising,

> **a multiplexer having a first input, a second input for receiving a 0 value, and a third input receiving a parameter designating a respective assembler code** and, depending on how the parameter is set, a different relative addressing takes place;
>
> a program counter;

a computation unit for computing relative addresses;

an adding unit connected between said program counter and said computation unit, said adding unit having a ~~first input connected to said program counter, a~~ second input connected to said multiplexer, and an output connected to said computation unit; and

a memory for storing an instruction length and having an output connected to said first input of said multiplexer. [emphasis added by Applicants]

Similarly, Applicants' claim 4 recites a microprocessor for processing various assembler codes, comprising,

a multiplexer having a first input, a second input for receiving a 0 value, and a third input receiving a parameter designating a respective assembler code and, depending on how the parameter is set, a different relative addressing takes place;

a program counter;

a computation unit for computing relative addresses;

an subtracting unit connected between said program counter and said computation unit for the relative addresses, said subtracting unit having a first input connected to said program counter, a second input connected to said multiplexer, and an output connected to said computation unit; and

a memory for storing an instruction length and having an output connected to said first input of said multiplexer. [emphasis added by Applicants]

With regard to Applicants' claim 3, item 10 of the Office Action, states that GOETZ fails to teach:

- **A multiplexer** having a first input a second input for receiving a zero value, a third input receiving

a parameter designating a respective assembler code, a memory for storing an instruction length having an output connected to said first input of said multiplexer

- An addition unit connected between said program counter and said computation unit, said adding unit having a first input connected to said program counter, a second input for an instruction length, and an output connected to said computation unit: [emphasis added by Applicants]

With regard to Applicants' claim 4, item 17 of the Office Action states that **GOETZ**, in view of **YOSHIDA** and **BORAR** (Applicants reiterate the above-noted confusion regarding **YOSHIDA** and **BORAR**) fails to teach the only limitation different from claim 3, which is a "subtracting unit" instead of an "adding unit".

The **YOSHIDA** reference discloses a branch address calculating system for branch instructions. The Office Action cited to the **YOSHIDA** reference as allegedly disclosing an x86-branch instruction that adds an offset to the address of the instruction following the branch instruction.

More particularly, the YOSHIDA reference discloses a method for calculating a target branch address when an instruction decoder has detected a branch within the instruction flow. See, col. 3 of YOSHIDA, lines 50 - 54. The object of YOSHIDA is to have the target branch address already calculated before the branch instruction reaches the execution step. See, col. 3 of YOSHIDA, lines 19 - 24. Thus, the methods and device disclosed in YOSHIDA only calculate the relative address of a machine code instruction. In this, YOSHIDA is analogous to Applicants' claimed "computation unit for computing a relative address". However, like GOETZ, YOSHIDA fails to teach or suggest, among other limitations, Applicants' particularly claimed multiplexer having a first input, a second input for receiving a zero value, a third input receiving a parameter designating a respective assembler code; a memory for storing an instruction length and having an output connected to said first input of said multiplexer; and Applicants' respectively claimed adding unit (claim 3)/ subtracting unit (claim 4) having a second input connected to said multiplexer. As stated in item 16 of the Office Action, no hardware implementation has been provided by YOSHIDA to describe how the word length value is generated.

The combination of GOETZ and YOSHIDA does not teach or suggest Applicants' particularly claimed circuit arrangement.

Further, the **MANO**, **POWERPC** and **SYSTEMS DESIGN** references do not cure the above-discussed deficiencies of **GOETZ** and **YOSHIDA** (i.e., those references do not disclose a **multiplexer**, particularly arranged as claimed by Applicants).

Rather, item 16 of the Office Action stated:

> Furthermore, since the combination of Goetz and Yoshida presents a system in which two instruction sets with two respective branch target address generation schemes are implemented, and in which a Q-bit (Goetz) is already used to indicate which instruction set addressing mode is to be used for a particular instruction, and no hardware implementation has been provided by Yoshida to describe how the "word length" value is generated, **one of ordinary skill in the art would have recognized to use a multiplexor to select which one of the two "word length" value types is needed (zero or variable) to present using the Q-bit as the selecting parameter** To further clarify, the combination of Goetz and Yoshida presents a problem of needing to select between two different values for the "word length", one being a zero for PowerPC-like branch instructions, and the other being the normal "word length" used for the x86-like branch instructions. **One of ordinary skill in the art would have recognized that a multiplexers** [sic] function is to select between two options using a controlling parameter (as evidenced by Section 3-7 of Logic and Computer Design Fundamentals, Kime), and since Goetz already teaches the Q-bit being used to indicate to hardware throughout the system which instruction set is currently being executed (x86 or PowerPC), it would have been obvious to use the Q-bit as the selecting parameter input to the multiplexer. Using the multiplexer would provide the advantage of solving the inherent problem of providing the correct "word length" to the 1st adder in figure 2, either a zero or the "word length" that would otherwise be provided when not executing PowerPC-like branch instructions. [emphasis added by Applicants]

Applicants' respectfully disagree that, absent hindsight reconstruction of the present invention, it would have been obvious to a person of ordinary skill in the art to: provide a multiplexer; provide the multiplexer having the specific inputs of Applicants' claims; and precisely arrange the adder/subtractor and memory circuit elements of Applicants' claims with the provided multiplexer. Rather, item 16 of the Office Action assumes a very lot. Nothing in the references teaches, or suggests providing a multiplexer that controls how a relative addressing takes place. Further, nothing in the cited references teaches or suggests connecting a multiplexer up in precisely the manner claimed by Applicants. The LOGIC reference merely comments on the standard operation of multiplexers, but does not disclose their configuration as claimed by Applicants.

The Patent Office has recognized that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion or motivation to do so found either in the references themselves, or in the knowledge generally available to one of skill in the art. The references cited herein clearly fail to teach, suggest or provide any motivation a person of skill in the art to provide a multiplexer as particularly configured in Applicants' claims 3

and 4, for use in the inventions of Applicants' claims 3 and 4. As such, Applicants' claims are believed not to be obvious over the cited art.

It is accordingly believed that none of the references, whether taken alone or in any combination, teach or suggest the features of claims 3 and 4. Claims 3 and 4 are, therefore, believed to be patentable over the art.

In view of the foregoing, reconsideration and allowance of claims 3 and 4 are solicited.

In the event the Examiner should still find any of the claims to be unpatentable, counsel would appreciate receiving a telephone call so that, if possible, patentable language can be worked out.

If an extension of time for this paper is required, petition for extension is herewith made.

Please charge any fees that might be due with respect to Sections 1.16 and 1.17 to the Deposit Account of Lerner Greenberg Stemer LLP, No. 12-1099.

Respectfully submitted,

Kerry P. Sisselman
Reg. No. 37,237

For Applicants

August 2, 2006

Lerner Greenberg Stemer LLP
Post Office Box 2480
Hollywood, FL 33022-2480
Tel: (954) 925-1100
Fax: (954) 925-1101